

BAB IV

PEMBUATAN ALAT

4.1 Pembuatan *Prototype*

Pada pembuatan Tugas Akhir ini diperlukan metode perancangan yang terstruktur dan sistematis untuk menghasilkan *prototype* sistem regenerasi otomatis *silica gel* pada *dehydrating breather* transformator yang dapat bekerja sesuai dengan tujuan perancangan. Bab ini membahas proses pembuatan alat secara keseluruhan, mulai dari perencanaan, pemilihan komponen, pembuatan perangkat keras (*hardware*), pembuatan perangkat lunak (*software*), hingga proses integrasi dan pengujian setiap komponen menjadi suatu sistem yang utuh. Setiap tahapan memiliki keterkaitan dalam menghasilkan sistem yang mampu mendeteksi kondisi *silica gel* berdasarkan perubahan warna, memantau suhu proses regenerasi, serta mengendalikan proses pemanasan secara otomatis. Secara umum, proses pembuatan alat dibagi menjadi dua tahapan utama, yaitu pembuatan perangkat keras (*hardware*) dan pembuatan perangkat lunak (*software*).

Pada pembuatan perangkat keras (*hardware*), tahap awal dilakukan dengan mengidentifikasi dan menentukan komponen-komponen yang sesuai dengan kebutuhan sistem. Komponen utama yang digunakan meliputi ESP32 sebagai mikrokontroler, sensor warna TCS3200 untuk mendeteksi perubahan warna *silica gel*, *thermocouple type - k* sebagai sensor suhu, dan modul MAX6675 sebagai antarmuka pembacaan *thermocouple* dengan ESP32. Sistem pemanas menggunakan *heater* yang dikendalikan melalui SCR Dimmer untuk mengatur besar daya pemanasan. Selain itu, digunakan relay sebagai sakelar pengendali *heater*, *power supply* sebagai sumber tegangan sistem kontrol, serta tabung atau ruang regenerasi sebagai tempat *silica gel* selama proses pengeringan. Seluruh komponen tersebut kemudian dirancang dan dirakit berdasarkan diagram blok dan rangkaian sistem yang telah dibuat. Pembuatan perangkat keras menjadi bagian penting dalam realisasi *prototype* karena keberhasilan proses regenerasi sangat bergantung pada integrasi sensor, sistem kendali, dan sistem pemanas yang tepat.

Tahap selanjutnya adalah pembuatan perangkat lunak (*software*) yang berfungsi untuk mengatur logika kerja sistem pada mikrokontroler ESP32. Program

dirancang untuk membaca data dari sensor TCS3200 dan *thermocouple type – k* melalui modul MAX6675. Sensor TCS3200 digunakan sebagai parameter untuk mengidentifikasi kondisi *silica gel* berdasarkan perubahan warna, sedangkan *thermocouple type – k* digunakan untuk memantau suhu selama proses regenerasi. Data dari kedua sensor diproses oleh ESP32 untuk menentukan kondisi sistem dan mengendalikan relay serta SCR Dimmer sebagai bagian dari pengendalian *heater*. Ketika *silica gel* terdeteksi telah jenuh berdasarkan parameter warna yang telah ditentukan, sistem akan mengaktifkan proses regenerasi dengan menyalakan *heater*. Selama proses tersebut, suhu dipantau secara terus-menerus untuk menjaga proses pemanasan tetap berada pada batas yang telah ditentukan. Apabila suhu mencapai batas maksimum atau kondisi warna *silica gel* telah menunjukkan bahwa proses regenerasi selesai, ESP32 akan memberikan perintah untuk menghentikan proses pemanasan.

Selain sistem kendali otomatis, *prototype* dilengkapi dengan konektivitas Wi-Fi yang terdapat pada ESP32 untuk mengirimkan data hasil pembacaan sensor ke platform Blynk. Data yang ditampilkan meliputi nilai suhu dari *thermocouple type – k*, kondisi warna *silica gel* berdasarkan pembacaan TCS3200, status *heater*, serta status proses regenerasi. Dengan adanya Blynk, kondisi *prototype* dapat dipantau secara *real-time* melalui perangkat yang terhubung dengan internet. Pengembangan program dilakukan menggunakan Arduino IDE, dengan struktur program yang meliputi inisialisasi sensor, pembacaan data, pengolahan kondisi *silica gel*, pengendalian *heater*, pengiriman data ke Blynk, dan pengamanan proses regenerasi. Integrasi antara perangkat keras dan perangkat lunak tersebut menghasilkan *prototype* yang mampu melakukan monitoring suhu dan warna *silica gel* serta proses regenerasi secara otomatis.

4.2 Alat dan Bahan

Alat dan bahan yang digunakan dalam pembuatan prototipe sistem regenerasi otomatis *silica gel* pada *dehydrating breather* transformator terdiri atas komponen elektronik, komponen mekanik, serta peralatan pendukung perakitan. Komponen elektronik digunakan untuk membangun sistem monitoring dan pengendalian, yang meliputi proses pembacaan suhu, pendeteksian warna *silica gel*,

pengendalian *heater* , serta komunikasi data ke aplikasi Blynk. Komponen mekanik berfungsi sebagai media pemasangan dan pelindung seluruh rangkaian agar sistem dapat beroperasi dengan aman dan rapi. Sementara itu, peralatan pendukung digunakan selama proses perakitan, seperti penyolderan, pengukuran tegangan, pengecekan sambungan, dan pemasangan komponen pada box panel. Berikut adalah tabel alat dan bahan yang diperlukan.

Tabel 4.1 Alat Pembuatan Tugas Akhir

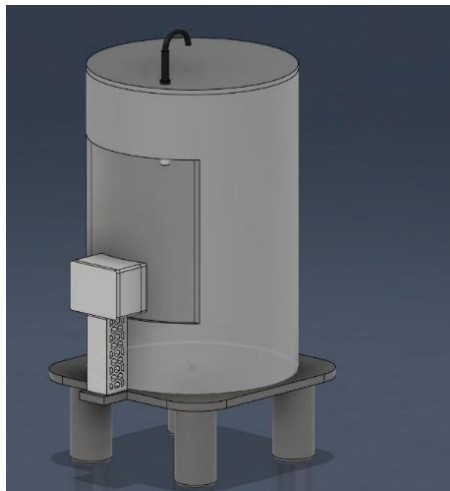
NO	NAMA	JUMLAH
1.	Multimeter	1
2.	Kabel USB	1
3.	Laptop	1
4.	Solder	1
5.	Timah Solder	1
6.	Tang Potong	1
7.	Tang Lancip	1
8.	Obeng	1
9.	Gerinda	1
10.	Cutter	1

Tabel 4.2 Bahan Pembuatan Tugas Akhir

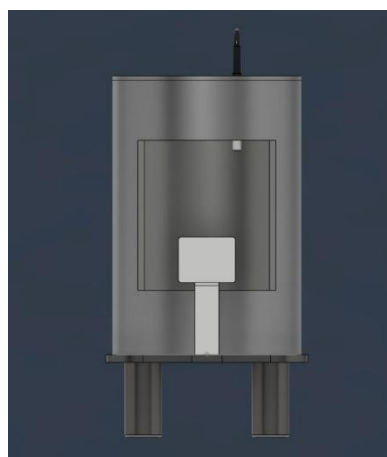
NO	NAMA	JUMLAH
1.	ESP 32 DEV-KIT 1	1
2.	Sensor Warna TCS3200	1
3.	Sensor Suhu <i>Thermocouple type - k</i>	1
4.	<i>Heater</i>	1
5.	LCD 20x4 with I2C	1
6.	Push Button NO	3
7.	Modul Step-Down 5v	1
8.	Modul Adaptor 12v	1
9.	Dimmer AC	1
10.	Transistor	3
11.	Modul Max 6675	1
12.	Kabel Steker AC	1
13.	Box Hitam X6	1
14.	JST XH 04P	3
15.	JST XH 02P	8
16.	Relay 220VAC	1
17.	Pilot Lamp 22mm 220VAC	3
18.	Swicth On/Off	1

4.3 Perancangan Mekanik

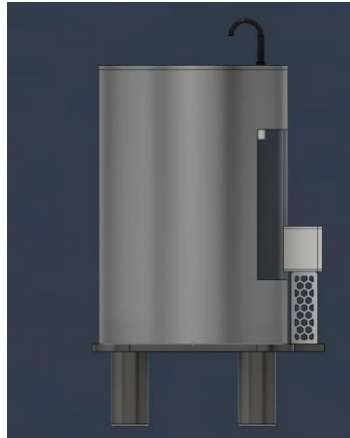
Pada sisi mekanik, perancangan diawali dengan pembuatan desain tiga dimensi menggunakan perangkat lunak CAD sebagai acuan dalam pembuatan prototipe sistem regenerasi *silica gel*. Desain pada gambar menunjukkan tabung regenerasi *silica gel* berbentuk silinder yang berfungsi sebagai tempat penyimpanan sekaligus proses pengeringan *silica gel*. Pada bagian atas tabung terdapat penutup yang dilengkapi dengan pegangan, sehingga penutup dapat dibuka dan ditutup dengan mudah ketika melakukan pengisian maupun pengambilan *silica gel*.



Gambar 4 .1 Kerangka Tangki Tampak Samping Kanan



Gambar 4 .2 Kerangka Tangki Tampak Depan



Gambar 4. 3 Kerangka Tangki Tampak Samping Kiri

Gambar 4.1, 4.2, 4.3 merupakan kerangka tangki tampak samping kanan, depan, dan kiri dari rancang bangun sistem regenerasi otomatis *silica gel* pada *dehydrating breather* menggunakan esp32 dan blynk. Pada bagian depan tabung terdapat jendela pengamatan yang berfungsi untuk melihat kondisi *silica gel* selama proses regenerasi. Pada bagian samping bawah terdapat kotak komponen yang digunakan sebagai tempat pemasangan sensor atau komponen pendukung sistem. Bagian bawah tabung dilengkapi dengan dudukan berbentuk lingkaran dan kaki penyangga yang berfungsi menjaga tabung tetap berdiri stabil serta memberikan jarak antara bagian bawah tabung dengan permukaan tempat alat diletakkan.

Secara keseluruhan, desain mekanik tersebut dibuat untuk menghasilkan prototipe yang kokoh, stabil, dan mudah dioperasikan. Bentuk tabung dipilih untuk menyediakan ruang regenerasi *silica gel*, sedangkan penempatan jendela pengamatan dan komponen pada bagian samping mempertimbangkan kemudahan dalam melakukan pemantauan dan pemeliharaan. Desain ini juga menjadi acuan dalam proses pembuatan prototipe agar setiap komponen mekanik dan elektronik dapat ditempatkan sesuai dengan fungsi serta kebutuhan sistem regenerasi otomatis *silica gel*.

4.4 Perancangan Elektrik

Pada perancangan elektrik sistem regenerasi *silica gel* pada transformator menggunakan ESP32 dan Blynk, dibuat sebuah *prototype* sebagai pusat

pengendalian, pembacaan sensor, dan pengoperasian *heater* . Rangkaian elektrik ditempatkan di dalam sebuah box kontrol agar komponen terlindungi serta susunan sistem menjadi lebih rapi dan terorganisasi. Pada bagian dalam box terlihat sebuah PCB sebagai pusat integrasi rangkaian, tempat ESP32 dan jalur koneksi berbagai komponen dihubungkan. Penggunaan PCB bertujuan untuk mengurangi penggunaan kabel secara berlebihan, membuat koneksi antarkomponen lebih terstruktur, serta memudahkan proses perakitan dan pemeliharaan *prototype*.



Gambar 4.4 Penempatan Komponen

Berdasarkan Gambar 4.4 merupakan penempatan komponen pada box panel pada rancang bangun sistem regenerasi otomatis *silica gel* pada *dehydrating breather* menggunakan esp32 dan blynk. ESP32 terlihat berada pada bagian bawah-tengah box dan terpasang pada PCB. ESP32 berfungsi sebagai mikrokontroler utama yang mengolah seluruh data masukan dan mengendalikan keluaran sistem. ESP32 menerima data dari sensor warna TCS3200 untuk mengetahui kondisi *silica gel* berdasarkan karakteristik warna, serta menerima data temperatur dari Thermocouple Type-K melalui modul MAX6675. Data tersebut kemudian diproses sesuai dengan program yang telah dibuat untuk menentukan kondisi sistem dan pengoperasian *heater* . ESP32 juga berfungsi sebagai penghubung sistem dengan jaringan Wi-Fi sehingga data dapat dikirimkan ke aplikasi Blynk dan Google Spreadsheet.

Pada bagian kiri box terlihat modul sensor warna yang dihubungkan menggunakan beberapa kabel ke rangkaian utama. Modul tersebut digunakan

sebagai antarmuka sensor TCS3200 yang berfungsi membaca karakteristik warna *silica gel*. Data warna yang diperoleh digunakan oleh ESP32 untuk menentukan kondisi *silica gel*, sehingga sensor ini menjadi salah satu bagian penting dalam menentukan kapan proses regenerasi perlu dilakukan dan kapan proses dapat dihentikan.

Di bagian atas-tengah box terdapat modul catu daya yang digunakan untuk menyediakan tegangan bagi rangkaian kontrol. Modul tersebut terhubung dengan jalur kabel menuju PCB dan komponen lainnya. Catu daya digunakan untuk menyuplai kebutuhan tegangan pada ESP32, sensor, relay, serta komponen pendukung lainnya. Pemusatan sumber catu daya pada bagian atas membuat distribusi kabel menuju rangkaian kontrol dapat ditata dengan lebih teratur.

Sementara itu, modul relay berwarna biru terlihat berada di atas ESP32. Relay berfungsi sebagai sakelar elektronik untuk mengendalikan penyalan dan pemutusan *heater*. ESP32 tidak menghubungkan *heater* secara langsung karena *heater* membutuhkan daya yang lebih besar dibandingkan kemampuan keluaran GPIO ESP32. Oleh karena itu, ESP32 memberikan sinyal kontrol kepada relay, kemudian relay melakukan proses switching pada rangkaian daya *heater*. Penempatan relay yang berdekatan dengan rangkaian kontrol juga memudahkan koneksi antara sinyal kendali dari ESP32 dengan rangkaian aktuator.

Pada bagian kanan dan bagian atas box terdapat komponen antarmuka yang digunakan sebagai bagian dari pengoperasian dan indikator sistem. Komponen-komponen tersebut terhubung ke rangkaian kontrol melalui kabel yang diarahkan menuju PCB. Keberadaan komponen antarmuka ini memungkinkan pengguna memberikan perintah pengoperasian sistem serta mengetahui kondisi kerja *prototype* secara langsung. Selain kontrol melalui perangkat keras, sistem juga dapat dipantau melalui Blynk.

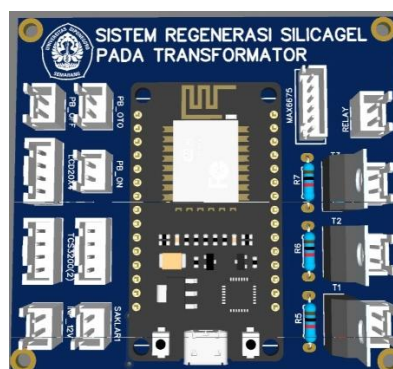
Pada bagian PCB juga terdapat jalur koneksi untuk modul MAX6675 dan *Thermocouple Type-K*, meskipun modul MAX6675 tidak terlihat secara jelas dari sisi tampilan gambar karena sebagian area tertutup oleh kabel. *Thermocouple Type-K* digunakan untuk mengukur suhu selama proses regenerasi, sedangkan MAX6675 berfungsi sebagai antarmuka yang mengubah sinyal *thermocouple* menjadi data

digital yang dapat dibaca oleh ESP32. Data suhu tersebut digunakan untuk memantau temperatur proses dan sebagai salah satu parameter pengaman agar *heater* tidak bekerja pada suhu yang melebihi batas yang telah ditentukan dalam program.

Susunan kabel pada *prototype* menunjukkan adanya pemisahan koneksi antara sensor, rangkaian kontrol, dan aktuator. Kabel dari sensor serta komponen antarmuka diarahkan menuju bagian PCB, sedangkan koneksi menuju relay dan sumber catu daya ditempatkan pada bagian atas dan tengah box. Pengaturan ini bertujuan agar jalur instalasi lebih terorganisasi dan mempermudah proses pemeriksaan apabila terjadi gangguan pada sistem.

Secara keseluruhan, *prototype* pada Gambar 4.1 menunjukkan bahwa PCB, ESP32, relay, modul catu daya, sensor warna, dan komponen antarmuka telah diintegrasikan di dalam satu box kontrol. ESP32 berperan sebagai pusat pengolahan data, TCS3200 digunakan untuk mendeteksi kondisi warna *silica gel*, Thermocouple Type-K dan MAX6675 digunakan untuk memantau temperatur, sedangkan relay digunakan untuk mengendalikan *heater*. Seluruh komponen tersebut bekerja secara terintegrasi dengan sistem monitoring Blynk dan Google Spreadsheet, sehingga proses regenerasi *silica gel* dapat dilakukan secara otomatis berdasarkan kondisi sensor dan dipantau oleh pengguna.

4.5 Perancangan PCB



Gambar 4 .5 Desain PCB

Gambar 4.5 merupakan desainPCB dari rancang bangun sistem regenerasi otomatis *silica gel* pada *dehydrating breather* menggunakan esp32 dan blynk.

Perancangan PCB (*Printed Circuit Board*) dilakukan untuk mengintegrasikan rangkaian elektronik pada sistem regenerasi otomatis *silica gel* pada *dehydrating breather* transformator menggunakan ESP32 dan Blynk menjadi satu papan rangkaian yang lebih ringkas, rapi, dan mudah digunakan. Gambar 4.1 merupakan desain PCB yang berfungsi sebagai pusat koneksi antara mikrokontroler, sensor, tombol kontrol, rangkaian driver, serta perangkat keluaran sistem.

Pada bagian tengah PCB terdapat ESP32 DevKit sebagai mikrokontroler utama yang berfungsi mengolah data dari sensor dan memberikan perintah kepada perangkat keluaran. Di bagian kiri PCB tersedia beberapa konektor untuk PB_ON, PB_OFF, PB_OTO, TCS3200(1), TCS3200(2), dan SAKLAR 12V. Konektor TCS3200 digunakan untuk menghubungkan dua sensor warna yang berfungsi mendeteksi kondisi *silica gel*, sedangkan push button digunakan sebagai kendali mode operasi sistem. Pada bagian kanan PCB terdapat konektor MAX6675 untuk menghubungkan *thermocouple type-k*, konektor relay, serta rangkaian transistor TIP122 (T1, T2, dan T3) yang digunakan sebagai driver untuk mengendalikan lampu indikator.

PCB juga dilengkapi dengan beberapa resistor R5, R6, dan R7 yang berfungsi sebagai pembatas arus pada basis transistor TIP122. Transistor tersebut digunakan sebagai sakelar elektronik agar keluaran ESP32 dapat mengendalikan lampu indikator dengan lebih aman. Konektor relay digunakan sebagai penghubung antara rangkaian kontrol dengan sistem pemanas, sehingga ESP32 dapat memberikan perintah untuk mengaktifkan atau mematikan *heater* selama proses regenerasi *silica gel*.

Selain itu, PCB menyediakan jalur dan konektor untuk catu daya 12 VDC yang digunakan sebagai sumber utama rangkaian. Tegangan tersebut selanjutnya disesuaikan dengan kebutuhan masing-masing komponen melalui rangkaian catu daya yang digunakan pada sistem. Penempatan konektor pada bagian tepi PCB bertujuan mempermudah pemasangan kabel sensor, push button, relay, dan sumber tegangan serta memudahkan proses perawatan apabila terjadi penggantian komponen.

Secara keseluruhan, desain PCB ini mengintegrasikan ESP32, MAX6675,

dua sensor TCS3200, tiga push button, relay, transistor TIP122, resistor, serta konektor catu daya dalam satu papan rangkaian. Perancangan tersebut bertujuan membuat sistem lebih ringkas, rapi, mudah dirakit, dan mudah dipelihara, sehingga dapat mendukung pengoperasian *prototype* regenerasi otomatis *silica gel* secara optimal.

4.7 Pembuatan Perangkat Lunak

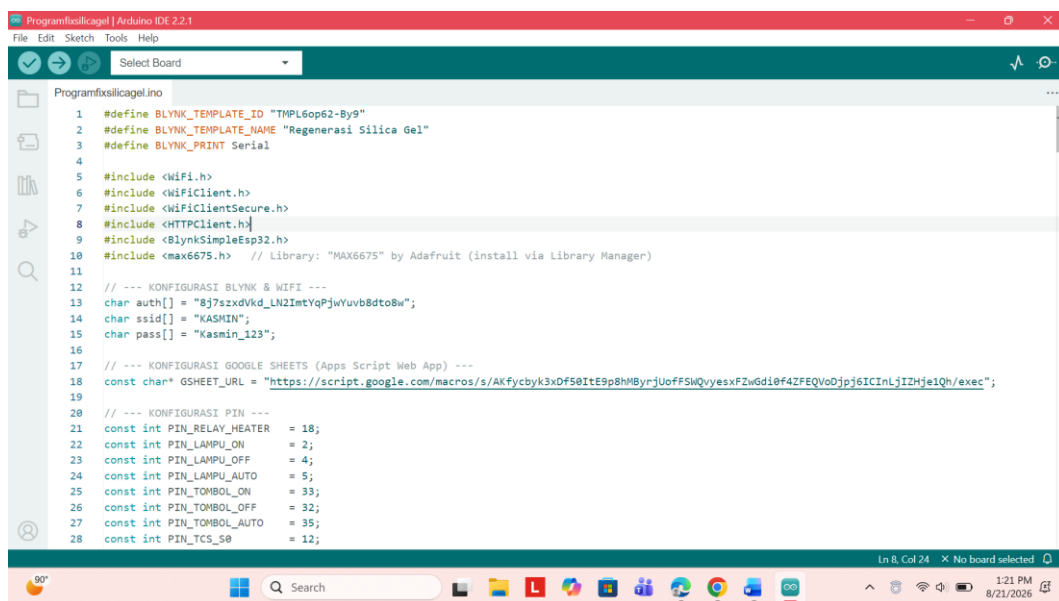
Setelah proses pembuatan dan perakitan perangkat keras selesai, tahap selanjutnya adalah pembuatan perangkat lunak (*software*) yang berfungsi sebagai pusat pengendalian sistem regenerasi otomatis *silica gel*. Pemrograman mikrokontroler dilakukan menggunakan *software* Arduino IDE dengan board ESP32 Dev Module. Board tersebut dipilih karena memiliki konektivitas Wi-Fi serta jumlah GPIO yang memadai untuk menghubungkan sensor, aktuator, dan komponen pendukung lainnya. Program dirancang untuk mengatur proses pembacaan sensor TCS3200 sebagai pendeteksi perubahan warna *silica gel* dan *thermocouple type -k* yang terhubung melalui modul MAX6675 sebagai pengukur suhu selama proses regenerasi.

Dalam pembuatan program digunakan beberapa *library* yang disesuaikan dengan komponen pada sistem, antara lain WiFi.h untuk menghubungkan ESP32 dengan jaringan internet, BlynkSimpleEsp32.h untuk komunikasi dan pengiriman data ke aplikasi Blynk, SPI.h untuk komunikasi antara ESP32 dengan modul MAX6675, serta library sensor TCS3200 untuk membaca nilai intensitas warna RGB. Data dari sensor kemudian diproses oleh ESP32 untuk menentukan kondisi *silica gel* dan mengendalikan relay serta SCR Dimmer yang digunakan untuk mengatur kerja *heater*. Ketika TCS3200 mendeteksi *silica gel* dalam kondisi jenuh berdasarkan parameter warna yang telah ditentukan, ESP32 menjalankan proses regenerasi dengan mengaktifkan *heater*. Selama proses berlangsung, *thermocouple type -k* secara terus-menerus mengukur suhu dan memberikan data kepada ESP32 sebagai parameter pengendalian. Data suhu, nilai warna, status *heater*, dan status regenerasi selanjutnya dikirimkan ke Blynk sehingga kondisi alat dapat dipantau secara *real-time*. Dengan demikian, program yang dibuat mengintegrasikan proses monitoring, pengendalian suhu, pendeteksian kondisi *silica gel*, dan regenerasi

otomatis dalam satu sistem berbasis ESP32

4.8 Inisialisasi Sistem

Tahap awal pembuatan perangkat lunak pada sistem regenerasi otomatis *silica gel* adalah inisialisasi sistem, yaitu proses mempersiapkan ESP32 agar dapat mengenali, mengatur, dan menjalankan seluruh komponen sesuai dengan fungsi yang telah dirancang. Berdasarkan program pada Gambar 4.6, inisialisasi diawali dengan pendefinisian Blynk Template ID dan Template Name serta penggunaan BLYNK_PRINT Serial untuk menampilkan informasi proses Blynk melalui Serial Monitor. Selanjutnya digunakan beberapa library, yaitu WiFi.h dan BlynkSimpleEsp32.h untuk koneksi jaringan dan Blynk, WiFiClient.h, WiFiClientSecure.h, dan HTTPClient.h untuk komunikasi dengan Google *Spreadsheet*, serta max6675.h untuk membaca *thermocouple type - k* melalui modul MAX6675.



Gambar 4 .6 Insialisasi Sistem

Gambar 4.6 merupakan inisialisasi sistem pada awalan rancang bangun sistem regenerasi otomatis *silica gel* pada *dehydrating breather* menggunakan esp32 dan blynk. Pada bagian berikutnya dilakukan konfigurasi Wi-Fi dan Blynk menggunakan auth, ssid, dan pass, sehingga ESP32 dapat terhubung ke jaringan internet dan mengirimkan data monitoring ke aplikasi Blynk. Sistem juga

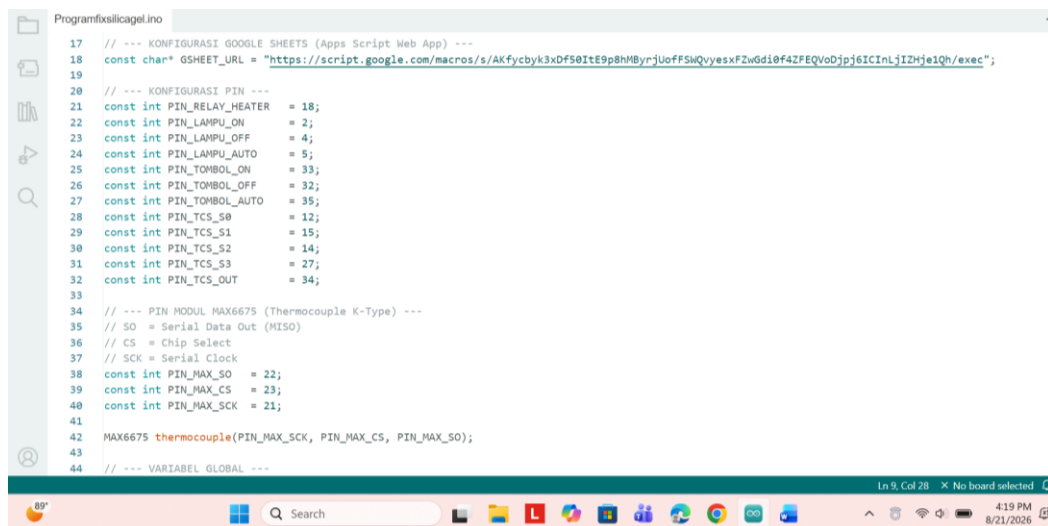
menginisialisasi alamat Google Apps Script Web App melalui variabel GSHEET_URL yang digunakan sebagai jalur pengiriman data ke Google *Spreadsheet*. Setelah komunikasi dikonfigurasi, dilakukan pendefinisian pin perangkat keras menggunakan const int, meliputi relay *heater* pada GPIO 18, lampu indikator ON pada GPIO 2, lampu indikator OFF pada GPIO 4, lampu indikator AUTO pada GPIO 5, push button ON pada GPIO 33, push button OFF pada GPIO 32, dan push button AUTO pada GPIO 35. Sensor warna TCS3200 juga dikonfigurasi melalui pin S0, S1, S2, S3, dan OUT, sedangkan pembacaan suhu dilakukan menggunakan *thermocouple type - k* yang terhubung ke modul MAX6675.

Setelah seluruh konfigurasi awal selesai, ESP32 menjalankan fungsi setup() untuk mengaktifkan komunikasi Serial, menghubungkan perangkat ke Wi-Fi dan Blynk, mengatur pin input dan output, serta memberikan kondisi awal pada relay dan sensor. Relay *heater* diatur dalam kondisi OFF pada saat sistem pertama kali dinyalakan untuk mencegah *heater* langsung aktif sebelum kondisi *silica gel* dibaca. Sensor TCS3200 kemudian dikonfigurasi untuk membaca komponen warna Red, Green, dan Blue (RGB), sedangkan MAX6675 diberikan waktu awal agar pembacaan thermocouple lebih stabil. Setelah proses inisialisasi selesai, ESP32 masuk ke fungsi loop() untuk menjalankan proses utama secara berulang, yaitu membaca suhu dan warna *silica gel*, menentukan kondisi *silica gel* Ungu (Basah) atau Biru (Kering), mengendalikan *heater* melalui relay, mengatur lampu indikator, serta mengirimkan data ke Blynk dan Google Spreadsheet.

4.8.1 Konfigurasi Sensor *Thermocouple Type-K* dan MAX6675

Salah satu komponen yang digunakan pada sistem yaitu *Thermocouple Type K* yang berfungsi sebagai parameter input untuk mengukur suhu pada proses regenerasi *silica gel*. Sensor ini digunakan untuk mengetahui perubahan suhu selama proses pemanasan sehingga kondisi temperatur di dalam sistem dapat dipantau dan digunakan sebagai salah satu parameter dalam pengendalian *heater*. Pada sistem, *Thermocouple Type K* dihubungkan dengan modul MAX6675 yang berfungsi sebagai antarmuka pembacaan sensor dengan ESP32. Untuk melakukan

inisialisasi sensor, digunakan library MAX6675 yang menyediakan fungsi untuk membaca data temperatur dari *Thermocouple Type K*. Komunikasi antara modul MAX6675 dan ESP32 dilakukan melalui tiga jalur, yaitu Serial Clock (SCK), Chip Select (CS), dan Serial Data Out (SO). Konfigurasi ini menjadi bagian penting dalam proses pembacaan suhu karena data temperatur dari suhu *thermocouple type -k* terlebih dahulu diproses oleh MAX6675 sebelum diterima oleh ESP32. Penjelasan lebih rinci mengenai inisialisasi perangkat lunak suhu *thermocouple type -k* dan MAX6675 disampaikan pada bagian berikut.



```

Programfixsilicagel.ino
17 // --- KONFIGURASI GOOGLE SHEETS (Apps Script Web App) ---
18 const char* GSHEET_URL = "https://script.google.com/macros/s/AkFycbyk3xDf50ItE9p8HMBYrjUoFF5wQvyesxFZwGdi0f4ZFEQVoDjpj6ICInLjIZHj1Qh/exec";
19
20 // --- KONFIGURASI PIN ---
21 const int PIN_RELAY_HEATER = 18;
22 const int PIN_LAMPU_ON = 2;
23 const int PIN_LAMPU_OFF = 4;
24 const int PIN_LAMPU_AUTO = 5;
25 const int PIN_TOMBOL_ON = 33;
26 const int PIN_TOMBOL_OFF = 32;
27 const int PIN_TOMBOL_AUTO = 35;
28 const int PIN_TCS_S0 = 12;
29 const int PIN_TCS_S1 = 15;
30 const int PIN_TCS_S2 = 14;
31 const int PIN_TCS_S3 = 27;
32 const int PIN_TCS_OUT = 34;
33
34 // --- PIN MODUL MAX6675 (Thermocouple K-Type) ---
35 // SO = Serial Data Out (MISO)
36 // CS = Chip Select
37 // SCK = Serial Clock
38 const int PIN_MAX_SO = 22;
39 const int PIN_MAX_CS = 23;
40 const int PIN_MAX_SCK = 21;
41
42 MAX6675 thermocouple(PIN_MAX_SCK, PIN_MAX_CS, PIN_MAX_SO);
43
44 // --- VARIABLE GLOBAL ---
  
```

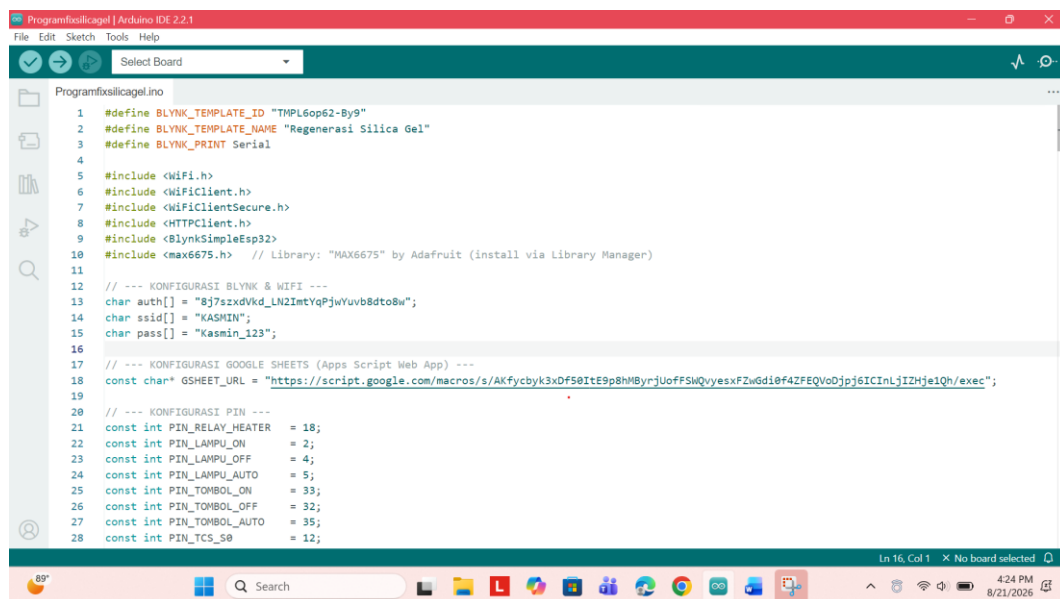
Gambar 4. 7 Konfigurasi Sensor *Thermocouple Type-K* dan MAX6675

Gambar 4.7 merupakan konfigurasi koding yang digunakan untuk inisialisasi *thermocouple type- k* dan modul MAX6675. Proses ini diawali dengan pendefinisian pin komunikasi yang digunakan oleh modul MAX6675, yaitu GPIO 22 sebagai Serial Data Out (SO), GPIO 23 sebagai Chip Select (CS), dan GPIO 21 sebagai Serial Clock (SCK). Pin SO berfungsi sebagai jalur keluaran data dari MAX6675 menuju ESP32, sedangkan pin CS digunakan untuk memilih modul MAX6675 pada saat proses pembacaan data dan pin SCK digunakan sebagai jalur clock komunikasi. Selanjutnya dilakukan pembuatan objek thermocouple dengan perintah `MAX6675 thermocouple(PIN_MAX_SCK, PIN_MAX_CS, PIN_MAX_SO);`. Melalui inisialisasi tersebut, ESP32 dapat berkomunikasi dengan modul MAX6675 dan memperoleh nilai suhu dari Thermocouple Type K. Data suhu yang diperoleh selanjutnya dapat digunakan untuk memantau temperatur

heater selama proses regenerasi *silica gel* serta menjadi dasar pengendalian sistem dan monitoring melalui Blynk.

4.8.2 Konfigurasi Wi-Fi dan Blynk

Salah satu bagian penting pada sistem yaitu konfigurasi koneksi WiFi dan Blynk yang berfungsi sebagai penghubung antara ESP32 dengan aplikasi monitoring. Blynk digunakan untuk memantau kondisi sistem regenerasi *silica gel*, seperti suhu dan status *heater* secara real-time melalui jaringan internet. Pada program, proses inisialisasi diawali dengan penggunaan beberapa library, yaitu WiFi.h untuk menyediakan fungsi koneksi WiFi pada ESP32, WiFiClient.h untuk komunikasi jaringan, WiFiClientSecure.h untuk komunikasi jaringan yang aman, HTTPClient.h untuk komunikasi menggunakan protokol HTTP, dan BlynkSimpleEsp32.h untuk menghubungkan ESP32 dengan platform Blynk. Penggunaan library tersebut menjadi dasar agar ESP32 dapat terhubung ke jaringan WiFi dan selanjutnya melakukan komunikasi dengan aplikasi Blynk.



```

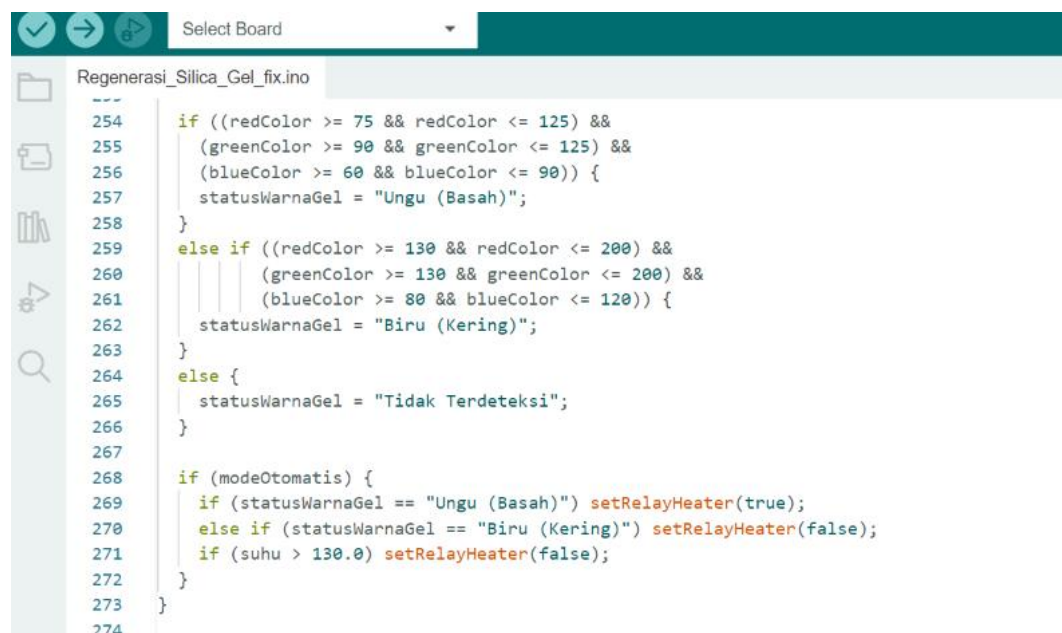
1 #define BLYNK_TEMPLATE_ID "TMPL6op62-By9"
2 #define BLYNK_TEMPLATE_NAME "Regenerasi Silica Gel"
3 #define BLYNK_PRINT Serial
4
5 #include <WiFi.h>
6 #include <WiFiClient.h>
7 #include <WiFiClientSecure.h>
8 #include <HTTPClient.h>
9 #include <BlynkSimpleEsp32>
10 #include <max6675.h> // Library: "MAX6675" by Adafruit (install via Library Manager)
11
12 // --- KONFIGURASI BLYNK & WIFI ---
13 char auth[] = "8j7szxdVkd_LN2ImtYqPjwYuvb8dto8w";
14 char ssid[] = "KASMIN";
15 char pass[] = "Kasmin_123";
16
17 // --- KONFIGURASI GOOGLE SHEETS (Apps Script Web App) ---
18 const char* GSHEET_URL = "https://script.google.com/macros/s/Akfcybyk3xDf50tE9p8HMByrjUoffF5WQyvesxFZw6d1eF4ZFEQVoDjpj6ICInLjIZHje1Qh/exec";
19
20 // --- KONFIGURASI PIN ---
21 const int PIN_RELAY_HEATER = 18;
22 const int PIN_LAMPU_ON = 2;
23 const int PIN_LAMPU_OFF = 4;
24 const int PIN_LAMPU_AUTO = 5;
25 const int PIN_TOMBOL_ON = 33;
26 const int PIN_TOMBOL_OFF = 32;
27 const int PIN_TOMBOL_AUTO = 35;
28 const int PIN_TCS_S0 = 12;
  
```

Gambar 4.8 Konfigurasi Wi-Fi dan Blynk

Gambar 4.8 merupakan konfigurasi koding yang digunakan untuk melakukan inisialisasi koneksi WiFi dan Blynk. Proses diawali dengan pendefinisian BLYNK_TEMPLATE_ID dan BLYNK_TEMPLATE_NAME yang digunakan untuk mengidentifikasi template perangkat pada platform Blynk. Selain

itu, terdapat BLYNK_PRINT Serial yang digunakan untuk menampilkan informasi proses komunikasi Blynk melalui Serial Monitor sehingga koneksi dapat dipantau saat proses pengujian. Selanjutnya, BlynkSimpleEsp32.h digunakan agar ESP32 dapat berkomunikasi dengan server Blynk. Pada bagian konfigurasi jaringan, terdapat variabel auth[] yang berisi Blynk Authentication Token, sedangkan ssid[] digunakan untuk menentukan nama jaringan WiFi yang akan digunakan dan pass[] digunakan untuk menyimpan kata sandi jaringan WiFi. Dengan konfigurasi tersebut, ESP32 dapat terhubung ke jaringan WiFi menggunakan SSID dan password yang telah ditentukan, kemudian melakukan autentikasi ke server Blynk menggunakan token yang tersedia. Setelah koneksi berhasil, ESP32 dapat mengirimkan data hasil pembacaan sensor dan status sistem ke aplikasi Blynk untuk keperluan monitoring proses regenerasi *silica gel* secara real-time.

4.8.3 Konfigurasi Sensor TCS3200



Gambar 4.9 Konfigurasi Sensor TCS3200

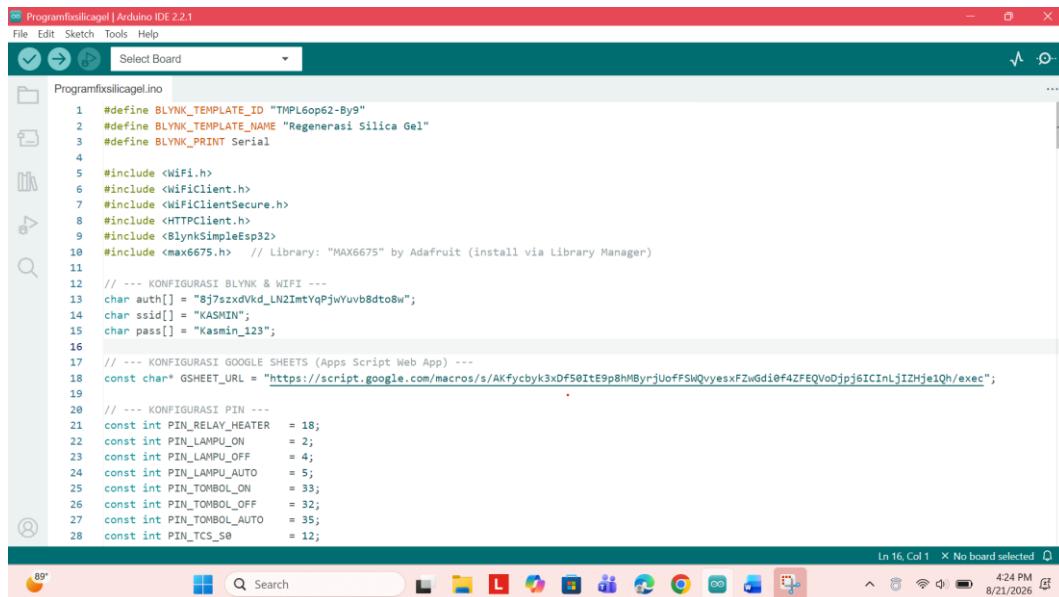
Gambar 4.9 merupakan konfigurasi sensor TCS3200 pada kodingan Rancang bangun sistem regenerasi otomatis . Sensor TCS3200 membaca tiga komponen warna, yaitu *Red*, *Green*, dan *Blue*, yang kemudian diproses oleh ESP32 dan disimpan dalam variabel redColor, greenColor, dan blueColor. Nilai tersebut selanjutnya dibandingkan dengan rentang hasil kalibrasi yang telah ditentukan pada

program. Pada program, nilai RGB tersebut dibandingkan dengan rentang hasil kalibrasi. kondisi ungu (basah) ditentukan apabila nilai R berada pada rentang 75–125, nilai G berada pada rentang 90–125, dan nilai B berada pada rentang 60–90. Sementara itu, kondisi biru (kering) ditentukan apabila nilai R berada pada rentang 130–200, nilai G berada pada rentang 130–200, dan nilai B berada pada rentang 80–120. Apabila nilai pembacaan berada di luar kedua rentang tersebut, sistem memberikan status “Tidak Terdeteksi”.

Hasil klasifikasi warna tersebut selanjutnya digunakan sebagai dasar pengendalian *heater* pada mode otomatis, yaitu *heater* diaktifkan ketika *silica gel* terdeteksi berwarna ungu dan dimatikan ketika *silica gel* berubah menjadi biru. Sistem juga dilengkapi batas keamanan suhu sebesar 130°C, sehingga *heater* akan dimatikan secara otomatis apabila suhu melebihi batas tersebut. Dengan konfigurasi tersebut, sensor TCS3200 dapat digunakan untuk mendukung proses regenerasi *silica gel* secara otomatis berdasarkan perubahan warna sebagai indikator kondisi *silica gel*.

4.8.4 Konfigurasi Google *Spreadsheet*

Selain menggunakan Blynk sebagai media monitoring, sistem juga dilengkapi dengan Google *Spreadsheet* yang digunakan sebagai media penyimpanan data hasil pengukuran. Data yang diperoleh dari sensor dapat dikirimkan dan dicatat ke dalam Google *Spreadsheet* sehingga hasil pengujian maupun proses regenerasi *silica gel* dapat terdokumentasi secara lebih terstruktur. Pada program, konfigurasi Google *Spreadsheet* dilakukan dengan mendefinisikan alamat Google Apps Script Web App yang berfungsi sebagai penghubung antara ESP32 dengan Google *Spreadsheet*.



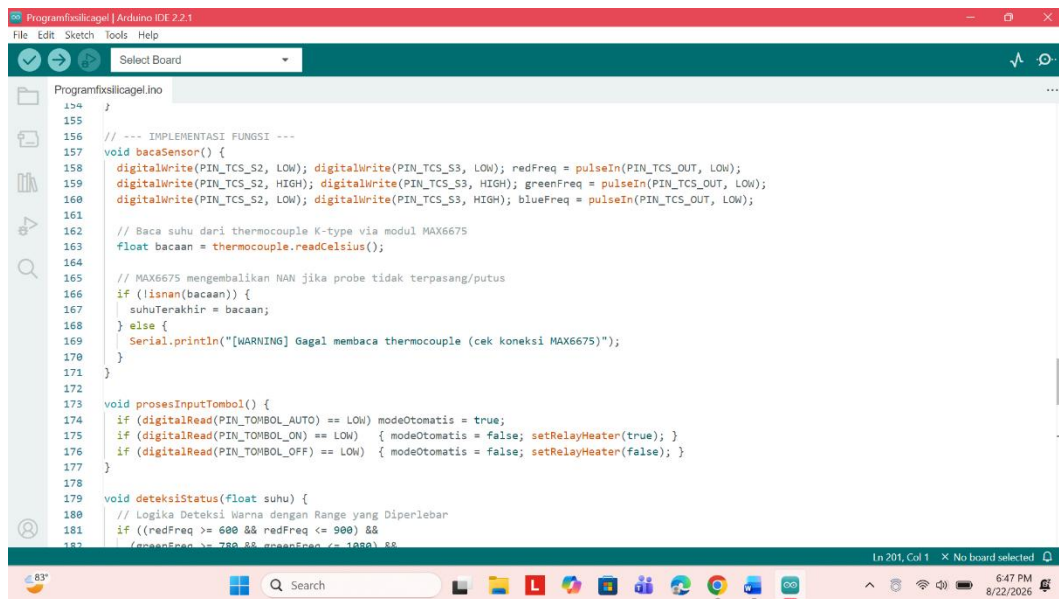
Gambar 4.10 Konfigurasi Google Spreadsheet

Gambar 4.10 merupakan konfigurasi koding yang digunakan untuk menghubungkan sistem dengan Google Spreadsheet. Proses ini dilakukan dengan mendefinisikan variabel `GSHEET_URL` menggunakan tipe data `const char*`. Variabel tersebut berisi alamat URL Google Apps Script Web App yang telah dibuat dan digunakan sebagai tujuan pengiriman data dari ESP32. Google Apps Script berfungsi sebagai perantara yang menerima data yang dikirimkan oleh ESP32 melalui koneksi internet, kemudian meneruskannya untuk disimpan ke dalam *Google Spreadsheet*. Dengan adanya konfigurasi tersebut, ESP32 dapat mengirimkan data hasil pembacaan sensor, seperti nilai suhu *thermocouple type-k* dan data sensor warna TCS3200, ke *Google Spreadsheet* secara otomatis. Data yang tersimpan selanjutnya dapat digunakan untuk dokumentasi, pemantauan, serta analisis hasil pengujian sistem regenerasi *silica gel*.

4.8.5 Konfigurasi Pembacaan Sensor dan Logika Input Sistem

Pada tahap pengoperasian sistem, ESP32 melakukan pembacaan data dari sensor TCS3200 dan *thermocouple type-k* melalui modul MAX6675. Sensor TCS3200 digunakan untuk mendeteksi perubahan warna *silica gel*, sedangkan Thermocouple Type-K digunakan untuk mengukur suhu selama proses regenerasi. Pembacaan kedua sensor dilakukan secara berkala sehingga kondisi *silica gel* dan

suhu di dalam sistem dapat dipantau oleh ESP32.



Gambar 4 .11 Konfigurasi Pembacaan Sensor dan Logika Input Sistem

Gambar 4.11 merupakan konfigurasi pembacaan sensor dan logika input sistem pada sistem .Pada sensor TCS3200, pembacaan warna dilakukan dengan mengatur kombinasi logika pada pin S2 dan S3. Kombinasi S2 = LOW dan S3 = LOW digunakan untuk membaca komponen warna merah (*Red*), kombinasi S2 = HIGH dan S3 = HIGH digunakan untuk membaca komponen hijau (*Green*), sedangkan kombinasi S2 = LOW dan S3 = HIGH digunakan untuk membaca komponen biru (*Blue*). Nilai masing-masing warna diperoleh melalui fungsi `pulseIn()` pada pin `PIN_TCS_OUT`. Nilai tersebut kemudian disimpan dalam variabel `redFreq`, `greenFreq`, dan `blueFreq` yang selanjutnya digunakan untuk menentukan kondisi *silica gel*.

Selanjutnya, pembacaan suhu dilakukan menggunakan *thermocouple type-k* yang terhubung dengan modul MAX6675. Perintah `thermocouple.readCelsius()` digunakan untuk memperoleh nilai suhu dalam satuan derajat Celsius. Nilai hasil pembacaan kemudian disimpan pada variabel `suhuTerakhir`. Program juga dilengkapi dengan pengecekan menggunakan `isnan()` untuk mengetahui apakah hasil pembacaan tidak valid, misalnya akibat thermocouple tidak terpasang atau terjadi gangguan komunikasi dengan modul MAX6675.

Selain pembacaan sensor, program juga melakukan pengolahan input dari

tiga tombol, yaitu AUTO, ON, dan OFF. Tombol AUTO digunakan untuk mengaktifkan mode otomatis, sedangkan tombol ON dan OFF digunakan untuk mengendalikan *heater* secara manual. Ketika tombol AUTO ditekan, variabel *modeOtomatis* diatur menjadi true. Apabila tombol ON ditekan, mode otomatis dinonaktifkan dan *heater* diaktifkan. Sebaliknya, tombol OFF akan menonaktifkan mode otomatis sekaligus mematikan *heater* .

Dengan demikian, bagian program pada gambar menunjukkan bahwa ESP32 menjadi pusat pengendalian sistem. ESP32 menerima data dari TCS3200 untuk mengetahui kondisi warna *silica gel* dan menerima data suhu dari *thermocouple type-k* melalui MAX6675. Data tersebut selanjutnya digunakan oleh program untuk menentukan kondisi *silica gel* serta mengendalikan *heater* melalui relay. Sistem juga memberikan pilihan mode manual dan otomatis, sehingga proses regenerasi dapat dilakukan secara otomatis berdasarkan kondisi *silica gel* dan batas suhu, maupun dikendalikan langsung oleh pengguna.